

Patent application of Bruce Middleton for:

**ZERO-FRONT-FOOTPRINT COMPACT INPUT SYSTEM**

**CROSS-REFERENCE TO RELATED APPLICATIONS**

This application is entitled to the benefit of Provisional Patent Application Ser # 60/442,327 filed 01/24/2003

and to the benefit of a Provisional Patent Application filed 11/19/2003, for which no file number has yet been received. (Title: Handset Input System, Express Mail item # ET 243585795).

**FIELD OF THE INVENTION**

This invention concerns input systems with alphanumeric capacity, which are compact enough for use on handheld and wearable devices, and more specifically such systems whose use does not occlude the front surface of the device.

**BACKGROUND OF THE INVENTION**

**COMPACT INPUT: THE STATE OF THE ART**

The emergence of portable information tools such as cell phones and handheld computers has created a need for extremely compact alphanumeric input systems which are nonetheless quick and convenient to use. The systems developed fall mostly into three categories: character-recognition systems, binary key arrays, and permutative systems.

Character-recognition systems have the benefit of exploiting pre-existing handwriting skills. However, they are intrinsically inefficient, because the inscription of characters is an inefficient form of data input. Analyzed as a trajectory, the average letter has five critical points, each of

which is an independent vector; thus inscription of a character is as physically complex as a sequence of five idealized keystrokes. As a result, handwriting speeds are about half as fast as the other systems considered here. Although once prominent in the form of Palm Computing's Graffiti™, their popularity has recently waned as people have come to appreciate that thumboards are faster and easier to use.

Binary key arrays can permit very high input rates for expert users, even in compact forms actuated by a stylus (most notably Fitaly™, from Textware Solutions, which uses a proprietary layout optimized for single-entry-point efficiency) or two thumbs (as popularized by the Blackberry™ from Research In Motion, which exploits the familiarity of the QWERTY layout). Indeed, contest results posted at the [www.Fitaly.com](http://www.Fitaly.com) website show speeds in excess of sixty words per minute for both of these systems. However, stylus-based systems involve the inconvenience of extracting the stylus and require two hands or a desktop, and thumboards are too large for contemporary phones. A recent thumboard – that of the Treo™ 600 (originally from Handspring Corporation and now from palmOne, Inc.) – refines the thumboard scrupulously, and achieves a width of 53 mm and a form factor which permits one-handed use. Even so, it cannot fit on contemporary cell phones. Some other recent innovations aim to sidestep the finger-width constraint keyboard size. Levy (US patent 5,973,621) uses disambiguation of multi-key strikes to permit sub-finger-width spacing. Another approach is to use fewer keys, but give each several functions; Lin et al [US patent 5,528,235], for example, use five-way keys in a phone keypad. A third approach – offered on most cell phones at present and thus the most widespread binary-key solution - is dictionary-based software disambiguation, most prominently T9™ from Tegic Corporation. It provides moderately fast input for skilled users entering text, but has several drawbacks - including the cognitive noise of the system's fluctuating guesses, the need to learn an arbitrary code (the assignment of letters to numeral keys), and the need to shift to some other system for dictionary-defying data such as numbers, names and Internet addresses.

Permutative systems use codes which associate sequences of input signals with generands. For alphanumeric capacity, pair codes are sufficient. The most lucid form is what may be termed the

“cellular menu”: an array of cells is displayed, each containing an array of symbols. A first directional signal selects a cell, and a second selects a symbol within the cell: a pair code is implicit in the menu. An early example is Knowlton (US patent 3,967,273 1976), who describes a telephone keyboard where each button is labeled with symbols representing its neighborhood; the keyboard itself functions as a cellular menu. Millington (US patent 6,256,029 2000) and Bozorgui-Nesbat (pending US application 20020136371) are more recent examples. These systems, however, involve two or three taps to generate a single character, which makes them uncompetitive in speed.

An improvement is possible by using strokes across a target matrix; this fuses a pair of taps into a single fluid gesture. Furthermore, the continuity of strokes provides a natural way of transmitting a gesture-termination or parsing signal: a break in contact signals gesture completion. Liljenwall et al (US patent 4,139,837 1979) and Curtin et al (US patent 5,521,986 1996) both describe systems of this sort, using the end points of traced trajectories as paired vector signals. Both, however, rely on unpromising mnemonic systems based on the graphical features of characters.

A system which combines stroke gestures with a cellular menu – and then adds an efficiency-optimized layout - is described in a second pending application (#20020136372) of Bozorgui-Nesbat, and implemented in his program MessagEase™ (ExIdeas Inc.). It has joined Fitaly and the Treo thumboard in achieving speeds of more than sixty words per minute in the Fitaly contest; together, these three systems define the current state of the art. It is possible to foresee at least two improvements which lie ahead for them: a clamshell handset with a thumboard will keep its weight securely within the user’s hand and eliminate competition for surface area between display and keyboard; and a system which combines the efficiency of Fitaly’s layout with the convenience and intrinsic speed of the Treo’s thumboard should be able to push the limits still further.

To summarize, several systems have shown that breakthrough performance is possible in a PDA form factor – but none of these have been successfully implemented even on the cell phones of today, with their small displays and relatively capacious front decks.

### **DEMANDS OF THE FUTURE**

For portable devices, progress brings two compelling imperatives into collision: devices shrink, and displays expand. The scope of PDA screens increases constantly, and has now passed the VGA threshold; phones are on the same track, though a couple of years behind. As the volume required for core functionality continues to shrink, the limiting factors which prevent phones from becoming the true personal computers increasingly become display size and intolerable input systems. The full-face display is inevitable – once people are trying to read documents, watch video and browse the Internet on their phones every square centimetre of display becomes precious. A related development is the computer watch – compellingly convenient, if it ever becomes easier to use than paper.

This is the logjam the current invention addresses: shattering some assumptions which keep input from fitting wherever it is needed. First, however, some prior art on a few related topics may be mentioned:

### **TENTATIVE SELECTION DISPLAY**

- Cziernecki (US patent 5,917,476 Cursor feedback text input method 1999) describes a system which uses a touch tablet as an unmarked keyboard: touching the tablet causes a character corresponding to the point of contact to be displayed, changing as the point of contact is changed; release of contact finalizes selection.
- Millington also provides tentative selection display, although at the cost of an added confirmation keystroke.

### **INPUT VIA NON-TRANSLATIONAL ANALOG INPUT DEVICE**

- Smith (US patent 5,982,203 1999) describes a permutative system marketed as Thumbscript, which uses angled trajectories. Although the marketed version has an array of binary sensors, Smith mentions use of a joystick as input device.
- Macor (US patent 5,901,222 1999) describes a telephone handset with a trackball or joystick in which all input is effected by activating virtual function keys.

### **BOTTOM-MOUNTED SENSOR**

- Bissett et al (US patent 5,543,588) describe a handheld computer using a bottom-mounted touchpad. This is however a purely hardware innovation, and does not touch on data entry.

### **SUMMARY OF THE INVENTION**

The practise of invention may be seen as the identification of false assumptions implicit in current practise, whose correction creates new freedom. The current invention aims to correct two false assumptions about input systems: that they need to be in a visible location so that the execution of input gestures may be visually supervised, and that keys make sense.

One basic objective is to migrate the actuatable elements of the input system away from the front surface of the device, so that the display can occupy the whole of that surface and be unobscured during use – without sacrificing the input rates, one-handed convenience and novice usability of the best current systems. This is possible because electronic technology makes it possible to monitor the input process more effectively than with direct visual supervision, by exploiting the very display whose growth is being enabled. A menu of input options – which provides all the benefits of a keyboard legend, and goes further in that it is never obscured by actuating digits, can be suppressed when not needed to permit more working display and can be reconfigured instantly to accurately show the options available in different modes - includes a live representation of the input signal the user is generating. To enable this display, a new gestural

style is adopted: transits are made while maintaining light normal pressure, and the sensor must be capable of distinguishing this from a firmer pressure exerted during gestures.

The second innovation is to decouple and redistribute various functions traditionally conflated in the concept of a “key”. A key is at once an actuatable element, a target, a visual feature, a tactile feature and (through its inscription) a menu element. The menu display shifts two of these functions away from the input-sensing system; the others also are better separated – which is made possible by using a single pointing device as the only primary input sensor. If keys are not used, clearances are no longer needed; finger width is eliminated as a parameter, and gestures may be optimally scaled to suit the precision with which humans can generate vectors. This reduces action, which will tend to increase speed and reduce stress and injury. The actuatable elements may be placed anywhere. For a handset, it is possible to utilize the most dexterous digits – index finger and thumb – and to place the actuatable elements within their natural range of motion while the handset is securely and comfortably gripped.

There are two implementations. One is a modified form of binary key array: targets correspond directly to generands, and are selected by simple pointing gestures. The other is a stroke-gesture, cellular-menu system.

In the pointing-gesture system, the actuatable element is a small touchpad, the targets are identifiable locations on the touchpad, and tactile features are a few ridges which suffice to identify targets and to establish continuous awareness of finger location. Scale of motion becomes arbitrary, and is reduced from the finger-width spacing of keyboards to the size of handwriting strokes – established over centuries as an optimal compromise between physical effort and the effort of concentration. The resulting system fits easily on the back of a phone, in the natural range of motion of the index finger. With the addition of a thumb-operable mode key on the side of the handset, an ample supply of signals – and other useful modes, such as a pointer mode and a panning mode (important for small devices) – are just a nudge away. (Such a mode

key has an advantage over standard shift keys, even on desktop keyboards, in that it is a parallel signal, generated by an otherwise-free digit.)

The cellular-menu / stroke-gesture system is implemented in both touchpad and isometric forms. The touchpad form is much like the binary-key-array implementation, but with a reduced scale of motion which may appeal to expert users. It uses the same hardware, so the user can be invited to toggle between the forms and perhaps migrate to the cellular form as skill develops. The isometric form eliminates finger motion altogether; as with notebook pointing devices, the choice between these two may be a matter of personal taste.

A watch implementation requires the extreme compactness of the cellular-menu form. It locates both the primary sensor and the shift key, not on the body of the watch, but on the strap buckle, which is more convenient for the other hand to access. This buckle may be stabilized by a secure grip between thumb and middle finger, with the signal threshold of the shift key set high enough to ignore the gripping force.

### **OBJECTS OF THE INVENTION**

The objects of the invention are to provide an input system which:

- is compact enough for use on pocketable devices such as cell phones and wearable devices such as watches;
- permits data entry by the hand holding the unit for handheld devices;
- uses gestures which are physically and cognitively simple, so that expert users can achieve high input rates;
- is instantly usable by novices;
- permits “eyes-free” operation by intermediate and expert users;
- permits the working display of the apparatus to cover its entire front surface;
- offers the output range of a full keyboard and a pointing-device;

- is cheap enough to be included as an interface device for any appliance or equipment which can benefit from character input or flexible control.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

Fig. 1 shows a possible menu with letters in alphabetic order.

Fig. 2 is a perspective view of a handset bearing a touchpad divided into three cells by tactile ridges

Fig. 3 is a front view of the handset of Fig. 3 with menu displayed

Fig. 4 is a perspective view of a handset bearing an array of discrete keys.

Fig. 5 is a perspective view of a watch showing primary sensor and shift key mounted on the watchstrap buckle.

## **DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS**

### **Fig. 1: gesture-to-generand mapping**

Figure 1 shows a cellular menu, which expresses a gesture-to-generand mapping. The choice of a mapping is not momentous if the system supports easy remapping. The technology provides a bonus here: the traditional inscribed keyboard serves as a chart which is virtually impossible to amend – but onscreen displays are perfectly malleable. They can show shift-mode generands when shift modes are in effect, and they can switch to different mappings as instantly and effortlessly as the interpretive system can. Plausibly desirable mappings include QWERTY on centre row and optimized single-entry-point mappings. Figure 1 shows another alternative: an alphabetic mapping. Fortunately, three of the most common letters – a, e, and i – receive single-target, “jab” gestures (as does the most common character of all, the space). As a result, the average vector count per character is 1.6, rather than 2. (Note that “□” denotes an unassigned gesture).

An alphabetic mapping may suit casual users: they need to find symbols on the menu as part of the input process, so an arrangement which makes this easy will be efficient for them. There is much room for debate on mappings. For example, highly efficient and organized mappings are possible if shift states are regarded as without penalty, and used freely: with separate tiers for numerals and math operators, symbols, and cursor/machine control, prime spots would be freed for letters. Fortunately, with no lock-in effect, it is possible for example to offer both a mapping optimized for reflex expert use and one optimized for menu-guided casual use as standard equipment, with users selecting one or the other with a tap to begin a session; and users can be given the ability to modify and define mappings. A legacy mode which assigns jabs to numerals may also be provided on phones. Foreign-language users, for example, will be free to download and swap menus and mappings as freely as they currently download ringtones; internationalization will be a solved problem.

To elaborate on the gestural system: there are four elements to a gesture – cell indication, confirmation (cell selection), symbol indication and confirmation (symbol selection). However, the gestural lexicon permits all four of these signals to be fused into a single stroke. Modulation of the planar components of the input variable effects navigation (and thus target indication), while modulation of the normal component effects selection. The user transits to the desired cell (with transited cells showing “indicated” highlighting as the cursor passes through them) while maintaining light normal pressure, confirms selection by increasing pressure above a value which serves as the threshold of the gesture range (at which point the highlighting of the selected cell changes to a form which means “selected”), and maintains pressure in the gesture range while selecting a symbol within the cell (with symbols similarly being highlighted as the cursor comes into their zones). In this second phase the scaling factor between input signal and cursor movement is reduced: fine motion on screen is produced without requiring finer input. Finally, reduction of normal pressure below the threshold value serves to confirm symbol selection; this terminates the gesture, and leaves the user ready to transit to the next cell target.

### **Gestural system and notation**

The gestural system divides the space of vectors generable by the pointing device into a meaningful zone (or gesture zone) subdivided into (preferably) nine target zones, and a non-meaningful (or transit) zone. In the preferred implementation, in which the Z variable is force, the transit zone is the region in which Z force is less than some meaningfulness threshold value  $Z_g$ : a uniform layer. Below this is the gesture zone. In the simplest case (where the second meaningful vector is planar location of termination point), this is divided into target zones by four lines which resemble a tic-tac-toe board and show positive and negative threshold values of X and Y components of the planar variable ( $F_x = \pm L_g$ ,  $F_y = \pm L_g$ ). Every gesture traces a trajectory through this vector space; the gesture begins when the trajectory passes through the threshold plane into the gesture zone, and ends when the trajectory passes back into the transit zone. For the touchpad implementation, strokes are real strokes through space; for the isometric implementation they are “force-strokes” through force-space.

Target zones may be identified by numeral in the pattern seen on a telephone keypad (1-3 on top, 4-6 as middle row, 7-9 on the bottom). A gesture is then specified by identifying the defining or critical points of a trajectory: 6, 15 and 2946 are possible gestures. In some cases intermediate zones will necessarily be crossed, but because they lie on the defined trajectory they are insignificant and are not mentioned: 123 is not a valid gesture, though in an extended gestural system which allowed for dwell time as a significant variable it might be.

“Jab” gestures are those in which only a single target zone is traversed. 3, for example, is the gesture for “i”, and 8 is the gesture for the space character. They may be interpreted as ordered pairs which repeat a single element. “Drag” gestures are those in which the trajectory passes through at least two target zones before returning to the transit zone. Thus 13 is the gesture for “c”, and 49 is the gesture for “9”. There are nine jabs and 72 simple drags: 81 single-stroke gestures. In distinguishing jabs from drags (for example, the cursor move 53 from the “e” gesture 3), the criterion is when  $Z_g$  is initiated: if before the pointing vector crosses out of the home target zone, the gesture is a drag; if after, it is a jab.

An example of a possible extended-gesture application is the treatment of the F1-F12 functions: they could be generated by spelling out their names (“f1” etc.) without releasing  $Z_g$  between characters. Although complex compared to other signals, the resulting trajectories are in the range of single inscribed characters: the longest have six critical points, while a capital “B” has eight.

As examples of possible extended gestural lexicons: an emphatic font (capital, bold, or italic, at the user’s choice) might be indicated by tracing a character stroke with Z force greater than some threshold value  $Z_2$ , well above normal levels; accents might be added by extending the stroke for a character with a gesture which draws the accent, executed above  $Z_2$  before the gesture is terminated. Going further, it is easy to provide programmability – permitting users to define their own gestures or their own gestural lexicons. Some possible uses: users could define the signature of their initials as a gesture (preferably with temporal sensitivity), then use it as a password which they will not forget, and which others cannot crack; and users familiar with Braille, shorthand or other symbolic systems would be free to develop mnemonically-resonant

implementations of them, without being constrained by a gestural lexicon chosen for other purposes.

Figures 2 and 3 show the pointing-gesture implementation, which uses an elongated touchpad 201 bearing tactile markings 202 which serve to identify a set of targets serving as effective keys. A suitable arrangement is a touchpad 30 mm wide, with tactile ridges serving to define three 10 mm squares. Each of nine zones within each square may serve as a key. (These zones are: 4 corners – fingertip firmly contacting two ridges; 4 sides – fingertip contacting one ridge – and centre – fingertip not firmly contacting any ridges. Both humans and touchpads have ample sensory resolution to distinguish these forms of actuation.). The mapping may follow qwerty convention, as shown by menu 301; mappings designed to optimize efficiency of single-digit input (such as the Fitaly and Opti keyboards) may also be used, as may an alphabetic ordering. The user may be given all of these options, and also the option of defining custom mappings. If a QWERTY layout is to be fit into nine columns, the tenth column may be accommodated by stroke gestures – a left-and-down stroke from the right-hand edge being a natural “Enter” gesture, while a straight-left gesture from the right edge makes sense for the backspace. P is the orphan – 303 shows a possible assignment, with the downward stroke from O being used for P. Given that there is only one anomaly, people will learn it quickly.

## Feedback

If there are multiple meaningful levels of normal force (or if it is desired to educate users about extravagant force levels, which are a well-documented ergonomic problem with keyboards), the feedback display can show the magnitude of the normal force, and threshold values for meaningful levels. This can be achieved in various ways without going beyond the idea of highlighting; for example, the background of an indicated cell or symbol can have a color which expresses normal force, either continuously or with abrupt changes at threshold values.

Better still is a live display of the input signal itself – shown as a trace across the menu, which serves as a map of input space. Learning is further facilitated by the fact that the system shows tentative selection of screen objects (both cells, in cellular systems, and symbols): input errors may be detected and corrected at each stage without generating output errors, and indication becomes selection only when a second signal is received.

In Fig 3, a trace 302 is shown which reveals the trajectory through input space since the last selection. Analog feedback must suit the gestural lexicon used. If strokes are defined by two locations, direct display of the trajectory suffices. However, if strokes are defined by initial target zone and direction what matters at each moment is not the absolute location of the contact point, but its displacement D (in input space) from the initial point of contact; the significant facts are the octant in which the direction of this vector lies, and whether its magnitude is greater than a noise threshold. A suitable system would draw a trace of the gesture, starting out at the centre of the contacted cell at initial contact. As in the minimal display option, the tentatively-selected symbol at each moment should be highlighted. Octant boundaries (and the circle around the cell centre which shows the noise threshold) could be shown, but are perhaps obvious enough in the jumps of the highlight from symbol to symbol as the cursor migrates.

Z-force may be represented by line width, and if so the gesture threshold can be shown as a pair of faint lines bracketing the trajectory. Color may also be used – for example, the trace can be green for transit range, yellow for recommended gesture range, red for excessive down-force.

Fortunately computer technology makes it possible to provide excellent visual feedback on gesture execution – better actually than direct sight of the gesturing digit – even though the operation is taking place in an occluded location. With an onscreen menu and an appropriate mapping rule, it is possible to give the user a live and precise display of the signal they are generating, in a context which also identifies the signals required to elicit generands and shows any difference between them. The user truly does *see what they are doing* – although they do not see themselves doing it.

### **Tentative selection display at cursor**

To encourage the user who is not quite certain of a gesture assignment to try out his or her best guess, the system can be programmed to display, at the cursor position, the symbol associated with any gesture after the gesture has been held for, say, half a second. (The pause prevents the visual noise of flashing symbols during normal input.) The user may then be given the options of

- releasing (which terminates the gesture and confirms the selection);
- aborting the selection by executing a *scrub* gesture (for example, a rapidly-reversed horizontal stroke);
- trying out other second targets by changing the signal vector;
- eliciting menu display by executing a *menu* gesture (for example, a rapidly-reversed vertical stroke).

This will be useful at an intermediate stage where the user knows the common characters and has varying degrees of certainty about others. Hunches can be painlessly tested if such feedback is provided; if not, guessing at associations will be inhibited and reliance on the menu will be extended.

Figure 4 shows a hybrid system, with a set of keys similar to a conventional phone keypad. As on some phones, these keys are wider than they are tall – about 10 mm wide and 4 mm tall. Each key may then be taken as offering three targets in an alphabetic mode: left edge, centre and right edge. If each key is mounted on two strain gauges separated laterally, pressure applied in each of these target regions will produce a distinctive pattern of strains. The alphabet may be mapped onto three rows of keys substantially in accordance with qwerty convention, or the alphabetic assignment used on phones may be adopted. The generands normally occupying the tenth column of a qwerty keyboard (comprising “P”, Backspace and Enter) may be assigned to keys on the fourth row, or to stroke gestures (since strokes across keys are distinguishable). This system is clearly less versatile than a touchpad and is not recommended, but is suggested as a possibly attractive intermediate form for some customers who might find an abrupt move to a keyless system too radical.

Figure 5 shows a watch with strap 503, bearing isometric sensor 502 on the face of buckle 501. Shift key 504 is located on a side of buckle 501; the buckle might be made reversible to accommodate right- and left-handed users. A small touchpad would also fit, at least on a man's watchstrap, but the pointing-gesture system cannot be accommodated. Applications such as this may justify the cellular-menu form, and once users have mastered it they may find it attractive even for devices which could fit a pointing-gesture sensor.

### Other envisioned applications

An option particularly relevant to women's watches, in which the need for display width clashes with a desire for a slender form, is to put the display on the inside of the bracelet, with the text orientation perpendicular to the axis of the arm. The sensor can then be on the back of the wrist, finger-operable, with a thumb-operable shift key on the side of the wrist. The display in this case will not be tall enough to show the menu all at once – but a viable system is possible. A display 8 mm tall can display two lines in a comfortable 9-point font, and three lines in 6-point (commonly used for "the fine print"). This scope can provide a partial option display, in a 3x3 matrix menu occupying one end of the display. Before a cell has been selected, the menu can show a 3x3 matrix representing the nine cells, with the jab generand for each displayed (one highlighted) and cell borders shown (like a tic-tac-toe board); initiation of  $Z_g$  selects a cell and brings up its symbol menu. To indicate where in the larger array the active cell is located, thick borders can be drawn around the cell on those sides which border neighboring cells: each cell has a unique set of such borders. To revoke a reconsidered cell selection,  $L_2$  can be used to push the cursor through a border; the menu then shows the symbols from the newly-selected cell. With diagonal pushes included, two will get you to any cell.

Such a reduced-display system can even be effected using a single character. In this case, the initial display can show a skeletal representation of the menu, with double lines for outer walls and single lines for inner walls; cell indication can still be shown by highlighting the indicated cell. Upon selection, the jab generand is shown, surrounded by walls on all sides – with the lines

dashed for those walls not adjacent to the symbol. The pattern of single and double lines identifies the cell, while the pattern of adjacencies identifies symbol position. Using L<sub>2</sub> jabs for cell jumps and permitting wrapping between cells and within cells, two jabs are enough to get anywhere in the matrix – so it's not hard to browse. Such diminutive displays might be used on tiny devices such as pen pagers or on very low-cost and possibly disposable items such as tickets.

A camera can put all controls on the force key, with a device-control mode. The range of gestures available permits camera operation to be simplified and enriched: for example, instead of requiring the user to select a shooting mode with a separate control and then redeploy the camera to shoot, lateral nudges on the force key can cause unit movements of a cursor across a set of mode symbols at the bottom of the rangefinder image while vertical force controls zoom and a firm inward push triggers capture. In advanced systems the user might toggle into a mode in which the force key controls a cross-hair cursor, moving over the range-finder image to designate a focusing target. The system can support multiple modes: for example a simple mode which offers only the controls so far described and minimizes the potential for error, and a more responsive advanced mode in which shooting-mode selection and zooming are effected by radial gestures with Z-force held below Z<sub>g</sub> (gestures which would be transit strokes in text mode) while jabs and drags can issue other commands (for example specifying aperture, exposure time and metering mode). At any time the user can toggle out of device-control mode to use various applications in which the force key follows normal character-input/cursor-control conventions – for example navigating and editing the portfolio of shots, fluently generating comments to be attached to images, and accessing the manual. Some commands within the device-control mode will be analog, but some may also incorporate characters generated by momentary reverersions to character-input mode. There is no reason to require a two-hand grasp of the device: the palm and three lower fingers can maintain a firm grip on the body while an index finger effects all control through the force key. One benefit of such simplicity is that the apparatus can fully accommodate left-handed users (which is seldom attempted at present) – even providing a second force key in a position convenient for left-handed use if necessary, at little cost.

An aftermarket version for existing handsets can attach a sensor to the lower surface, with a cable or infrared link to the system such as aftermarket keyboards use at present. The sensor may be attached with adhesives, or by use of a mechanical clamping fixture (the "sleds" which provide card slots for PDAs at present being one model); in the latter case the shift key can easily be accommodated on the side of the unit.

Other uses extend to every sort of product which can benefit from character input or flexible control. For devices such as camcorders which intrinsically have displays, the addition of a sensor makes it possible to provide a manual on a ROM chip; such a manual can never be lost, is always there when needed and can be searched instantly with one hand.